EnterCriticalSection

Thread termination or critical section deletion can cause undefined state

Sean Barnum, Cigital, Inc. [vita[1]]

2007-03-22

Part "Original Cigital Coding Rule in XML"

Mime-type: text/xml, size: 6033 bytes

Attack Category	• Denial of Service
Vulnerability Category	• Race Condition • Unhandled Exception
Software Context	• Critical Sections • Threads and Processes
Location	• winbase.h
Description	The EnterCriticalSection function waits for ownership of the specified critical section object. The function returns when the calling thread is granted ownership. If a thread terminates while it has ownership of a critical section, the state of the critical section is undefined. If a critical section is deleted while it is still owned, the state of the threads waiting for ownership of the deleted critical section is undefined.
APIs	**Function Name** / **Comments** EnterCriticalSection InitializeCriticalSection
Method of Attack	
Exception Criteria	
Solutions	**Solution Applicability** / **Solution Description** / **Solution Efficacy** Generally applicable. / To reduce impact of undefined states of critical sections and threads, ensure through path analysis that entry into a critical section / Somewhat effective.

1. http://buildsecurityin.us-cert.gov/bsi-rules/35-BSI.html (Barnum, Sean)

		is followed by an exit from the critical section.	
	If you think a thread may die while holding a critical section.	Use a mutex, which has abandonment semantics.	Somewhat effective.
	Generally applicable.	To reduce likelihood of deadlock scenarios, ensure that "time spent" in the critical section is as small as possible.	Somewhat effective.
	On Windows 2000/NT: In low memory situations	Use structured exception handling, or call the InitializeCriticalSectionAndSpinCount function to preallocate the event used by EnterCriticalSection instead of calling the InitializeCriticalSection function, which forces EnterCriticalSection to allocate the event.	This is only effective in more optimized use of memory.

Signature Details	void EnterCriticalSection(LPCRITICAL_SECTION lpCriticalSection);

Examples of Incorrect Code	`// Global variable`
	`CRITICAL_SECTION CriticalSection;`
	`void main()`
	`{`
	`...`
	`// Initialize the critical section one time only.`
	`if (! InitializeCriticalSectionAndSpinCount(&Criti 0x80000400) )`
	`return;`
	`...`
	`// Release resources used by the critical section object.`

```
DeleteCriticalSection(&CriticalSection)
}

DWORD WINAPI ThreadProc( LPVOID
lpParameter )
{
[...]

// Request ownership of the
critical section.
EnterCriticalSection(&CriticalSection);

// Access the shared resource.

// Release ownership of the
critical section.
// If for whatever reason this
call is omitted, then as noted in
the description, behavior
// is undefined.
//
LeaveCriticalSection(&CriticalSection);

[...]
}
```

Examples of Corrected Code

```
// Global variable
CRITICAL_SECTION CriticalSection;

void main()
{
...

// Initialize the critical section
one time only.
if (!
InitializeCriticalSectionAndSpinCount(&Criti
0x80000400) )
return;
...

// Release resources used by the
critical section object.
DeleteCriticalSection(&CriticalSection)
}

DWORD WINAPI ThreadProc( LPVOID
lpParameter )
{
[...]

// Request ownership of the
critical section.
EnterCriticalSection(&CriticalSection);

// Access the shared resource.

// Release ownership of the
critical section.
LeaveCriticalSection(&CriticalSection);
```

	`[...]` `}`
Source Reference	http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dllproc/base/entercriticalsection.asp[2]
Recommended Resource	
Discriminant Set	**Operating System**: • Windows **Languages**: • C • C++

Cigital, Inc. Copyright

The Build Security In (BSI) portal is sponsored by the U.S. Department of Homeland Security (DHS), National Cyber Security Division. The Software Engineering Institute (SEI) develops and operates BSI. DHS funding supports the publishing of all site content.

1. mailto:copyright@cigital.com